METHOD AND SYSTEM FOR NETWORK-BASED, DISTRIBUTED, REAL-TIME COMMAND AND CONTROL OF AN ENTERPRISE

BACKGROUND OF THE INVENTION

[0001] The present invention relates to business or enterprise management, and more particularly to methods and systems for managing value and assets of systems of enterprise systems.

[0002] There are a wide variety of existing business management and evaluation methods and systems. Significant efforts have been made in developing systems to measure financial performance, process and product quality, customer support, regulatory compliance, information systems availability, safety and security, supply chain, and other parameters and areas. However, most methods and systems, whether computerized or not, generally focus on a single aspect or domain of an enterprise (for example, supply chain management). Further, many of the methods and systems are constructed such that the evaluation or monitoring of the enterprise is performed from the outside looking in. For example, stock and market analysts may evaluate a company, but they do so from an outsider position and generally use information based on past events.

SUMMARY OF THE INVENTION

[0003] Using one or more single-aspect or domain-specific measurement systems to manage an enterprise has several deficiencies. A large, complex enterprise may use multiple measurement systems, and if an accurate overall picture of the enterprise is to be obtained, some combination or correlation of the information from each domain-specific measurement system must be accomplished. For example, it may be necessary to define the relationships between each domain-specific measurement system, rationalize their metrics, weigh and balance their respective values, resolve contradictions, and manage the cost of implementing and maintaining adherence to those systems.

[0004] Accordingly, there is a need for improved methods and systems for managing or controlling an enterprise that are useful to managers running the company, that are based upon real-time information, and that may be applied to or cover multiple aspects of the enterprise.

[0005] In one embodiment, the present invention provides a method of creating an enterprise control architecture. The method includes establishing five echelons of control, a first echelon, a second echelon, a third echelon, a fourth echelon, and a fifth echelon. Each echelon has one or more objects. The first echelon has an object that encapsulates one or more production processes, the second echelon has an object that provides control over the production process, the third echelon has an object that coordinates processes executed at the first echelon in light of enterprise objectives, the fourth echelon has an object that provides planning and development functions, and the fifth echelon has an object that provides supervisory control and that determines the enterprise objectives. The method also includes connecting each of the five echelons with a plurality of control links.

[0006] In some embodiments, this method also includes configuring each object of the first echelon such that each object may include an information port, configuring the third echelon to include an object that audits performance of processes at the first echelon, dividing an enterprise into multiple levels and, for each level, establishing five echelons of control. The method may also include dividing an enterprise of systems into multiples levels, and for each level, establishing five echelons of control.

[0007] In other embodiments, the invention provides a method of creating an enterprise control architecture. The method includes dividing a system into multiple levels, and for each level, establishing five echelons of control - a first echelon, a second echelon, a third echelon, a fourth echelon, and a fifth echelon. Each echelon has one or more objects. The first echelon has an object that encapsulates a production process, the second echelon has an object that provides control over the production process, the third echelon has an object that coordinates processes executed at the first echelon in light of enterprise objectives, the fourth echelon has an object that provides planning and development functions, the fifth echelon has an

object that provides supervisory control and that determines the enterprise objectives. The method also includes configuring each object of the first echelon such that each object may include an information port; configuring the third echelon to include an object that audits performance of processes at the first echelon; and connecting each of the five echelons with a plurality of control links.

[0008] This second method may be modified such that the first echelon has an object that encapsulates a supply chain process and a second object that encapsulates an asset chain process. The first object may operate according to a first transform function. And, the second object may operate according to a second transform function. Together these two objects comprise a value production unit ("VPU").

[0009] In other embodiments, the invention provides an enterprise control system having a plurality of value production units ("VPUs") connected in an addressable grid. Each production unit has a first echelon, a second echelon, a third echelon, a fourth echelon, and a fifth echelon. Each echelon has one or more objects. The first echelon has an object that encapsulates a production process and that includes an information port. The second echelon has an object that provides control over the production process, the third echelon has an object that coordinates processes executed at the first echelon in light of enterprise objectives and an object that audits performance of processes at the first echelon, the fourth echelon has an object that provides planning and development functions, and the fifth echelon has an object that provides supervisory control and that determines the enterprise objectives.

[0010] The enterprise control system also includes a plurality of control links connecting each of the five echelons. In some embodiments, the system also includes a router configured to control communications between at least some of the plurality of value production units.

[0011] In another embodiment, the invention provides a method of network-based, real-time command and control of systems of enterprise systems. The method includes providing a communications network; providing an interface for connecting to the network; providing an application interface for connecting to an enterprise

application; providing one or more value production units, each value production unit having four full-duplex ports; providing a router to dynamically create connections between the one or more value production units; providing one or more enterprise process controls, at least some of the one or more enterprise process controls coupled to at least some of the one or more value production units; and providing at least one enterprise management interface.

[0012] In another embodiment, the invention provides a system of network-based, real-time command and control of an enterprise. The system includes an enterprise operating system having an interface layer, a performance measurement layer, a process control layer, and a performance management layer; and one or more VPUs. Each VPU has four full-duplex ports and is interfaced with the performance measurement layer of the enterprise operating system. The system also includes a router to dynamically create connections between the one or more value production units.

[0013] In another embodiment, the invention provides a system for controlling an enterprise. The system includes a plurality of enterprise units, each enterprise unit having a first echelon including at least two objects, each object configured to execute a production process, a second echelon including at least one object to control one of the production processes of the first echelon, a third echelon including an object to coordinate processes based on objectives and available shared assets, a fourth echelon including an object to provide planning and development, and a fifth echelon having an object to set the objectives of the enterprise unit. The system also includes a potentiality measurement tool coupled to the fourth echelon; a capability measurement tool coupled to the third echelon; an actuality measurement tool coupled to the first echelon; and a performance metrics engine coupled to the performance measurement tool, the capability measurement tools, and the actuality measurement tool.

[0014] In another embodiment, the invention provides an enterprise operating system that includes a network interface layer configured to support one or more virtual machine services and one or more application interfaces; a performance

measurement layer configured to support one or more value production processes; a process control layer configured to support one or more supervisory processes; and a management interface layer configured to support one or more enterprise management interfaces.

[0015] In another embodiment, the invention provides an enterprise control architecture that includes a graphical user interface ("GUI"), or "bridge" configured to generate graphical information readable by a human and to generate command messages based on inputs from a human; a modeler coupled to the bridge; a plurality of multi-level production units coupled to a router; and a command parser coupled to the bridge and the router and operable to extract individual messages intended for specific ones of the plurality of multi-level production units from the command messages.

[0016] The architecture also includes an operations interface operable to receive raw data from selected levels of the plurality of multi-level production units; an operations data acquisition service coupled to the operations interface, the operations data acquisition service operable to deliver raw data to a data store; a data filter coupled to the operations data acquisition service, the data filter operable to process the raw data to generate processed data and to deliver the processed data to a data base; and a performance measurement engine coupled to the data filter and the modeler. The performance measurement engine is operable to generate performance metrics.

[0017] An alarms and events engine is coupled to the performance measurement engine and operable to generate alarm messages and event messages based on performance metrics received from the performance measurement engine. A history engine is coupled to the performance measurement engine and the alarms and events engine. The history engine is operable to log alarms and events in a data store. The alarms and events are based on the alarm messages, event messages, performance metrics, or combination of the same. A report generator is coupled to the history engine and operable to generate reports based on the alarms and events generated by the history engine. A display generator is coupled to the history engine, the report

generator, and the alarms and events engine. An icon engine is coupled to the modeler and the display generator. Finally, a display is coupled to the display generator.

[0018] In yet another embodiment, the invention provides a graphical interface for an enterprise control system. The graphical interface includes a center dash board with a first set of links to a plurality of control panels, each control panel configured to display a representation of a single level of a multi-level business unit; and a second set of links to a plurality of portals, the plurality of portals including an investor portal, customer portal, a supplier portal, and a subordinate portal.

[0019] Other features and advantages of the invention will become apparent to those skilled in the art upon review of the following detailed description, claims, and drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

[0020] In the drawings:

[0021] Fig. 1 is a schematic of logical components used in embodiments of the invention and their dependency on one another.

[0022] Fig. 2 illustrates a networked group of controllable process elements in enterprises governed by accountability hierarchies.

[0023] Fig. 3 illustrates an enterprise control model of one embodiment of the invention.

[0024] Fig. 4 illustrates an exemplary enterprise structure having five levels and applied to both a military and a commercial enterprise.

[0025] Fig. 5 illustrates an exemplary recursive control structure of a business area of an enterprise.

[0026] Fig. 6 illustrates end-to-end timing paths in a command structure.

[0027] Fig. 7 is a graphical representation of behavior in a system of enterprise systems.

[0028] Fig. 8 is an illustration of value production units in an asset-chain and supply-chain grid.

[0029] Fig. 9 is an illustration of how a value production unit may represent a node in a system of systems.

[0030] Fig. 10 is an illustration of how VPU's communicate within a lattice of systems of systems ("SOS") using a feedback mechanism in the form of an intra-lattice router.

[0031] Fig. 11 is a first model of VPU behavior.

[0032] Fig. 12 is a depiction of transport functions for value production units in the first model shown in Fig. 11, where the transport function describe the mathematical relations in the first model, emphasizing production flows.

[0033] Fig. 13 is an illustration of a second model of VPU behavior or dynamics, emphasizing financial flows.

[0034] Fig. 14 is a graph of relationships among system performance indices.

[0035] Fig. 15 is an illustration of how control of value production is provided by elements of a VPU controller and measurement services for a VPU.

[0036] Fig. 16 is an illustration of a measurement and control framework applied to the first and second models of value production.

[0037] Fig. 17A is a composite illustration of a performance measurement framework.

[0038] Fig. 17B is an illustration of performance metrics applied to five echelons or levels of an enterprise.

[0039] Figs. 18A & 18B are illustrations of a performance measurement framework for a recursive set of value production units.

[0040] Fig. 19 is an illustration of a management system implemented using teachings of embodiments of the invention showing how information from measurement feeds is processed and how commands from a bridge are directed to components of an enterprise.

[0041] Fig. 20 is an illustration of the major elements of an enterprise operating system according to one embodiment of the invention.

[0042] Fig. 21 is an illustration (in the form of a five-ring Smith chart) of enterprise applications that may be interfaced with an enterprise operating system.

[0043] Fig. 22 is an illustration of exemplary enterprise management interface bridge displays that may be generated by embodiments of the invention.

[0044] Fig. 23 is an illustration of how bridge displays may be used to track enterprise behaviors.

[0045] Fig. 24 is an illustration of the layering of core elements of an exemplary enterprise operating system.

[0046] Fig. 25 is an illustration of an exemplary enterprise performance management interface.

[0047] Fig. 26 is an illustration of an exemplary supervisory interface for the second model of a VPU.

[0048] It is to be understood that the invention is not limited in its application to the details of construction and the arrangement of components set forth in the following description or illustrated in the drawings. The invention is capable of other embodiments and of being practiced or of being carried out in various ways. Also, it is to be understood that the phraseology and terminology used herein is for the purpose of description and should not be regarded as limiting. The use of "including," "comprising," or "having" and variations thereof herein is meant to

encompass the items listed thereafter and equivalents thereof as well as additional items. Unless limited otherwise, the terms "connected," "coupled," and "mounted," and variations thereof herein are used broadly and encompass direct and indirect connections, couplings, and mountings. In addition, the terms "connected" and "coupled" and variations thereof are not restricted to physical or mechanical connections or couplings.

DETAILED DESCRIPTION

[0049] Before describing embodiments of the invention in detail, definitions of certain terms are provided.

[0050] As used herein, an "enterprise" is defined as an arbitrary interactive unit of an organization (noun) or work (verb) for systematically creating measurable value through the delivery of a product or service. ("Value" is defined below.) An enterprise can range in size from small-scale manufacturing cells (e.g., bio-chemical or electro-mechanical) in a production line to cooperation among large-scale national or international, public or private sector entities. Enterprises may interact with one another through or be involved in development and exploitation of value chains. The value chains may include *supply chains* along which products and services are consumed and produced, and *asset chains* along which investment assets are produced and consumed.

[0051] "Investments" may take the form of capital, matter, or energy. Enterprises that effectively participate in supply and asset chains rely on well-defined interfaces through which information (data) and control (execution threads) pass between and among enterprise neighbors along the two chains.

[0052] The core or nuclei of an enterprise that is responsible for creating and sustaining its value proposition(s) is referred to herein as a value production unit ("VPU"). A viable enterprise, then, is a continuous and sustainable computation of one or more value propositions within its contained VPU objects. In this context, "enterprise engineering" is the science and discipline of designing, deploying,

adapting, and maintaining federations of VPUs. "Enterprise management and control" is the process (profession) of governing such federations.

[0053] "Enterprise value production" is the act of converting assets (e.g., men and material) into returns, and of effectively utilizing these assets to meet customer demands for goods and services.

[0054] An enterprise (including its systems, processes, and threads) operates in "real-time" to the extent that timeliness is an intrinsic aspect of its correct behavior. Therefore, an enterprise, regardless of size, operates in real-time to the extent that it meets its timing (e.g., deadline) requirements. Operating in real-time is not the same as operating "on-line," or "seen through a web page," or operating "real fast." Timing issues embodied in requirements for *deadlines, response times, timeframes,* or *time constraints* are typically application-dependent. They are not simply functions of bureaucratic latencies, network bandwidths, processor speeds, or which browser, server, or network programming language one uses to create a man-machine or user interface. (Although these things may impact process timing, they do not provide methods to actually manage the resources needed to meet timing requirements.)

[0055] A system is "distributed" to the extent that its execution (e.g., threads, transactions, and messages) must pass through, or is required to complete one or more tasks within multiple "nodes." A "node" is defined as a uniquely identifiable or named (e.g., with an IP address) computational object. An enterprise consisting of joint military services, regionally deployed divisions, and collections of theater assets is multi-node. Likewise, a commercial enterprise comprising business areas, business units, production plants, and production units is multi-node. Management command and control decisions that must engage resources at two or more nodes are multi-node. A distributed enterprise is real-time to the extent that its management controls must meet end-to-end timeliness requirements as they propagate from node to node. Meeting such end-to-end timing requirements requires distributed real-time resource management policies and mechanisms.

[0056] "Timeliness" is a measure of two aspects of an enterprise or object: first, how well time constraints are met, individually and in ensemble; and, second, how well one can successfully predict meeting those constraints. In accordance with the teachings herein, it is preferred that a real-time enterprise provide its internal and external clients (e.g., suppliers, investors, customers, and subordinates) with means to express time constraints for the execution of specific tasks (e.g., project duration or completion milestone date, or order fulfillment date). Deadlines are a familiar, albeit simple, example of a time constraint. Other examples are provided herein.

[0057] "Predictability" is the ability to plan; to know *á priori*, with a specified level of certainty, the degree to which a system will meet its timing requirements. The predictability of an enterprise may depend on effective "resource management." "Resource management" may involve partitioning complex systems and consigning resources and their administration to semi-autonomous entities within an enterprise. Realizing a real-time enterprise generally requires objectifying its resources, effectively automating and distributing resource management, and requiring distributed resources to actively assist in meeting application end-to-end timeliness requirements.

[0058] "End-to-end timeliness" is the acceptable execution time of the data and applications logic in a multi-node system. Achieving end-to-end timeliness may require obtaining *á priori* service level agreements ("SLAs") from participating nodes so they can plan to meet the agreements. Achieving end-to-end timeliness may also require dynamic resource management (e.g., in response to failure) within each node to make best efforts to meet the time constraints under current conditions, or to assist downstream nodes with information useful in re-planning.

[0059] "Current conditions" are statements about the capacity available on a given node (i.e., its instantaneous capability) to meet its SLA commitments given the status of available resources (raw materials, processing power, manpower, etc.). The extent to which a multi-node enterprise system can achieve its end-to-end timing obligations is dependent, at least in part, on the resources and resource management policies that are available at each participating node at any given time. Since each node in a

sequence may introduce statistical variation in its ability to meet its obligations (e.g., completion times that are sometimes late, sometimes early, or sometimes fail), SLAs and corresponding execution time status should be propagated to node resource managers to ensure proper operation of certain embodiments of the invention.

[0060] A "federation" is two or more freely cooperating enterprises.

[0061] As used in connection with "enterprise organization," the term "organization" refers to static accountability hierarchies typically used in referring to enterprise command structure for managing production units (or processes) in production areas, areas in plants, plants in business units, business units in business areas (divisions), and business areas in corporations. "Levels of control," on the other hand, refers to the dynamic structure of an enterprise, the manner and means of acquiring and then administering valuable resources throughout an enterprise on behalf of distributed production objectives.

[0062] In relation to enterprise operations, the concept of "control" has several connotations. Loosely defined it means the management or regulation of a process or set of correlated processes. Control may be further classified by its degree of automation and by the degree of its independence, or conversely, its role in a collaborative framework.

[0063] Control activities include behavior generation and final control actuation. Control is the means by which the process under control is driven to its next-state conditions.

[0064] "Command and Control" or "C2" are the policies and mechanisms for exercising real-time authority and direction over interconnected and interdependent assets through a set of protocols and shared value propositions, while functioning in a collaborative and interactive (or network-centric) community governed by a formal hierarchical accountability structure.

[0065] The function of C2 of a given process is to sense (measure) the process parameters; assimilate and assess those measurements in the context of history and

current process states; update existing process models; generate appropriate next-state control behaviors; and issue commands to process actuators (final control devices). In addition, for intelligent control systems, the C2 model includes a function called value judgment that supports adaptive controls capable of adjusting default sensory perception and behavior generation capabilities.

[0066] "Measurement" generally entails the activation of sensors appropriate to the task of determining the present state of a process. Typically, there are sensors for various parameters, and often multiple sensors for the same class of measurement. The result of measurement is a data set containing records of the form:

```
measurement = {
        sensor_id,
        measurement_value,
        measurement_time_stamp,
        measurement_quality
        }
```

[0067] Sensors may operate synchronously (e.g., polled) or asynchronously (e.g., publish-subscribe) with respect to the sensory perception processes that lead to timely behavior generation and subsequent control. Measurement systems must operate in timeframes that correspond to the basic cyclic behavior of the processes under control. A basic engineering principle (the Nyquist Principle) dictates that, for process observability and controllability, measurement-sampling rates must be at least twice the fundamental frequency of the process.

[0068] "Situation assessment" is the process of assimilating the process measurements in order to determine the current state of the process. This may entail filtering, smoothing, and parameter estimation of the data sets. It may require the correlation of several data sets in order to determine the quality of the data sets themselves, perhaps adding compensation to sensor data to correct sensor errors.

[0069] "Planning and execution" contains model building and behavior generation activities, and for intelligent control systems, value judgment services. This set of activities is responsible for manipulating the policies and mechanisms that directly affect asset utilization. This is the domain of three echelons (discussed below): 5-4-3.

[0070] As it relates to certain models and methods presented herein, "value" is defined as the difference between the marginal cost (C_s) incurred by the supplier of a product or service and the benefit (B_c) perceived by its consumer, or

$$V_{sc} = B_c - C_s.$$

[0071] Thus, value is a relative measure, defining a gradient or potential difference between the two participants. This gradient, once above a certain minimum threshold (V_{sc}^{min}), is sufficient to power the flow of goods and services in one direction, and compensation (e.g., barter, money, etc.) in the opposite direction.

[0072] The volume of this flow is proportional to the stability of this gradient over time. Stable potentials allow the two participants to establish internal processes capable of sustaining (or regulating) the flows to meet their other operating requirements. Sustainable flows are an object of VPUs in achieving and maintaining homeostasis or dynamic equilibrium, which is generally required for viability.

[0073] "Value production" is a process, and as such it is governed by policies and procedures, depends on available fixed production assets, requires availability of consumable resources (i.e., raw materials), and produces by-products (e.g., side effects or waste). The process of value creation is typically distributed, involving participation from elements within and among cooperating systems. Furthermore, within a given system there may be many such processes, each addressing a different set of goals and objectives, some strategic, some operational, and some tactical. It is extended and assumed that within a given federation the core value production processes can be identified. The model of value production used in many embodiments of the invention is a VPU.

[0074] In any given system, value production takes place in each of three primary types of activities: strategic processes, operational processes, and tactical processes.

Strategic processes involve long-term goals and objectives. Non-existent or poor-quality strategic initiatives can undermine the vitality and viability of value production. Operational processes involve the effective execution of goals and plans and coordination of tactical processes. Poor operational performance can halt or reverse value creation. Tactical processes involve actual value production through production processes.

[0075] Fig. 1 illustrates a simplified architecture of a multi-level command and control or management system 5 that may be implemented using the teachings herein. The system 5 has a first or enterprise control level 7 that is based upon an enterprise control model or "ecm"). As will be discussed in greater detail below, the ecm is a recursive (scale-free) model that may be used throughout an enterprise and in a system of enterprises. An enterprise controller built according to the ecm includes five echelons: 1 (production process controller), 2 (regulatory controller), 3* (audit controller), 3 (operations controller), 4 (development (planning) controller), and 5 (supervisory controller).

[0076] The system 5 includes a second or enterprise valuation level 9 that depends on the enterprise control level 7. The enterprise valuation level provides the definition of a VPU (discussed in greater detail below). The system 5 also includes a performance measurement level 11 that provides multiple performance measurement tools (discussed below). Three other levels define the remainder of the system 5: an operating system level 13, a enterprise command and control level 15, and an enterprise workbench level 17. As indicated in the drawings, each subsequently higher level depends on the level below it (for example Level 15 depends on level 13).

[0077] Fig. 2 illustrates a group of controllable processes 25 arranged in a hierarchy with six levels ($L_0 - L_5$). At each level are unfolding trajectories 27 of the processes 25, including histories left of a center axis 29, and forward plans to the right of the axis 29. The model in Fig. 2 addresses levels of control and time: two aspects that while known are not addressed in most command and control management systems adequately or at all.

[0078] As will be discussed in greater detail below, the command and control methods and systems disclosed provide management user interfaces, or an "enterprise bridge." The term comes from the role the bridge provides as a focus of command and control for the captain and officers of large ocean-going vessels, a role similar to that of the cockpit or flight deck in an aircraft or spaceship. The idea is that management and control is most effectively exercised when the human elements are immersed in an interactive environment providing real-time, integrated, and context dependent automation and control systems. Certain of the exemplary command and control methods and systems disclosed provide

1. Nested, enterprise-operational-system elements that span low-level activities (L_0) up through high-level, enterprise-wide activities (L_5);

2. Data acquisition (or measurement feeds) from grid-connected or networked enterprise information systems;

3. Data analysis (or filtering) based on context and objectives;

4. Performance measurement tools (or index functions) for evaluating individual and composite performance of enterprise elements in a context neutral (or generalized) way;

5. Historian services for following behavior as a function of time, to support simulation and analysis, and to allow for tracking of event and sequence causality;

6. Report generators for periodic and event-driven documentation;

7. Enterprise system modelers for simulating scenarios and analyzing performance; and

8. Interactive user interfaces for the highest level of command of each process, including the process supervisor (e.g., commander), process developer (e.g., planner), and process operator (e.g., XO). The user interfaces are driven by a display generator that dynamically, based on

context, provides a shared "knowledge wall" and individual C2 displays for the three senior process officers.

[0079] Fig. 3 illustrates a control model 30 for a value production system. There are two major features of the control model 30: entities (classes or objects) of value creation 32 and an architecture of adaptive decision and control that is represented, in part, by connections or links 34 (which may represent regulatory control loops) between the entities 32. Taken together these two features establish requirements for an enterprise operating system ("EOS"), discussed in greater detail below. The EOS provides a distributed real-time execution environment (i.e., virtual enterprise machine) for hosting command and control applications.

[0080] Fig. 4 illustrates a hypothetical multi-divisional enterprise structure 36 having five organizational levels 38. A given enterprise may be more or less complex and have more or less levels. Further, the levels may be referred to using different terminology. Fig. 3 contrasts terminology used in the U.S. military's global command system ("GCS") architecture and terminology used in a typical commercial or private business hierarchy.

[0081] Each level 38 serves to define the resources (production policies and assets) it encapsulates, its peer entities, the sources of its investments (authority) above, and its subordinate entities below. In the example provided, policy and assets flow downward, and returns and assets deployed (or results) flow upwards. Net value production, however, is not directly represented, being the byproduct of a complex and dynamic set of relationships carried out throughout the hierarchy.

[0082] As suggested above, value chains typically comprise a web of relationships among internal and external entities that reside at different levels in their respective enterprises. Furthermore, such webs are often fluid, dynamically established and destroyed as situations evolve, as customer demand comes and goes, and technologies evolve, or as assets are produced and consumed. Accordingly, an enterprise organization has two characters – one static, one dynamic. The structure 36 depicts only the static organization of an enterprise.

[0083] To characterize an enterprise's dynamic structure, embodiments of the invention rely on classes, or echelons, of control. Classes or echelons of control are regulatory (i.e., reflexive feedback) mechanisms needed to ally and coordinate VPUs that participate in distributed real-time computations. Fig. 3 depicts five echelons of control (1, 2, 3, 4, and 5) in the enterprise control model 30. As depicted, the model 30 has three production processes (P1, P2, and P3). Each echelon of control has one or more associated classes or objects (E1, E2, E3, E4, and E5 or generically an entity 32). There is also an object E3*, which will be discussed below.

[0084] In the example illustrated, each of the three E1 classes or objects encapsulates one of the production process P1, P2, and P3. The E2 objects represent process controllers or directors. The E3 object represents the operations directorate responsible for coordination of the processes *vis-à-vis* overall enterprise objectives and available shared assets. The E3 object acts as an autonomic control center that provides a source of homeostasis. The E4 object provides planning and development functions, coupling volition above to autonomic behavior below, and striving to move the enterprise forward as a whole, guiding the allocation of its strategic assets between operational and development imperatives. The E5 object represents the governing or executive "board" functions, providing supervisory or "conscious" control (volition) over the enterprise.

[0085] In the embodiment shown, the enterprise control model 30 has its foundation in industrial dynamics and management cybernetics. These fields are generally directed to defining an enterprise as an observable and controllable process. Efforts based on these principles resulted in the introduction of a model of survivable ("viable") systems. Research into viable systems has focused on lessons from natural systems, and their organization and mechanisms for learning and adaptation in evolving contexts.

[0086] According to the viable system model ("VSM"), planning on the basis of actuality is "programming." Planning on the basis of capability is "operations," as is management by objectives ("MBO"). Planning on the basis of potentiality is "strategic," or normative. In this sense, controller classes E1 and E2 provide

programming; classes E3 and E3* provide operations; and classes E5 and E4 provide normative controls. E1 objects are directors (a.k.a., commanders or mangers) of the processes under control ("PUC"). Processes are where the work of the VPU (or entity 32) is done, and where autonomy should exist, because the processes generally require adaptation and reactivity to the environments they serve. As a consequence, the management at echelon 1 is important to the local success of the system. The collective performance of E1 processes constitutes the actual performance of the system.

[0087] E2 objects are generally needed when there is more than one E1 object. E2 objects provide a means of synchronizing multiple parallel processes. Synchronization deals with coordination of shared resources, and with prevention of oscillatory or deadlocked behaviors. E2 controls, are therefore, regulatory in nature, and exist outside of self-serving E1 prerogatives. As noted, E3 objects are focused on operations, the execution of current plans, and the management of resources shared among the E1 processes. E3 objects are responsible for achieving the current level of capability of a system.

[0088] The E3* echelon or object provides a semi-independent audit function for E3 operations. An audit function provides a process neutral assessment of the actual performance of the E1 processes as an aid to the E3 object in its interrelated roles of managing capability and efficient utilization of resources. The E4 planning and development class is responsible for looking at the environment within which the E3 class (and its E1 processes) operates. The E4 class also develops policies and mechanisms for the continuous improvement of the system. The E4 class provides future "what-if" analyses in an attempt at reprogramming the E3-E2-E1 complex. The E4 class is the regulator of change (or adaptation) in the system.

[0089] The E5 class acts as a supervisory controller responsible for the overall mission and associated policies (doctrines) which set the goals and objectives of the enterprise. The E5 class provides the end-point for alarms and events that cannot be resolved by the E3 class in synchronizing the E1 processes. The E5 class is also the final authority for changes proposed by the E4 class.

[0090] As part of the control system, and as noted above, each E2 class regulates a corresponding process (P1, P2, and P3). For example, the processes might represent three manufacturing plants within a business unit. In this example, objects E1 – E2 would then be the management staffs for each plant. Connections between E1 and E2 signify both a direct communication path as well as a specific control protocol. The goal of this control loop is local homeostasis. In addition to its E2 director, each E1 process participates in the E3 object's attempts at maintaining organic homeostasis. This may be accomplished through two antagonistic feedback control loops: the E3 object's "sympathetic" and E3 object's "parasympathetic" systems. The sympathetic system, as in animal physiology, is a reflex arc responsible for detecting sensory stimuli and generating qualified motor responses. The parasympathetic system provides an audit loop that serves to dampen high-gain processes' tendency to over-react to stimulus. It is the contention between these two control loops that enables viable systems to operate far from equilibrium, yet remain stable and highly responsive.

[0091] Fig. 5 illustrates a recursive control structure of a business area 40 of an enterprise with three business units 41, 42, and 43. Business area 40 is recursively defined in terms of its three business units (41, 42, and 43). The business units are in turn defined in terms of their embedded production areas, and so on. (For the sake simplicity further levels of detail are not shown.) The multiple tiers shown in Fig. 5 illustrate the recursive relationship among the five echelons within each enterprise domain; and how they participate as elements in viable systems below the business area 40 level (it should be understood that there could be elements above the business area 40 level too.) Thus, viable systems are coupled in two dimensions, horizontally across and vertically within the enterprise. This recursive structure works for military enterprises, as well as non-military enterprises such as healthcare providers, educational systems, and so on.

[0092] As will be explained in further detail below, the five echelon or multiple level architecture illustrated in Fig. 5 can be integrated with the VPU construct. More information regarding the integration of the VPU construct and the multi-tier or multi-

level business unit or component model will be provided in the discussion of Fig. 16 (below).

[0093] Fig. 6 illustrates command threads 55 and 57 as they wend their way through the command structure. The performance of the threads 55 and 57 effects enterprise dynamics, and is therefore important to enterprise management. As used herein threads are defined by the invocation of resources as they are brought to bear on the demands of a system. Threads may have both horizontal and vertical dimensions in the sense that they may follow resources within echelons of one level of an enterprise and may flow from one level to another. Proper performance of threads help ensure end-to-end timeliness, end-to-end reliability, continuity of nested relations, and federation control.

[0094] In summary, the business unit model proposed and illustrated in Figs. 3, 4, 5, and 6 separates the conscious (E5-E4) controls from the more autonomic (E3-E2-E1) controls of an enterprise. The linkage at E4-E3 provides a router or router-like mechanism that filters and switches commands and assets flowing downwards and operational information flowing upwards.

[0095] The implementation of a C2 system for managing behavior in the enterprise using the model shown in Fig. 3 requires a platform, or virtual machine, on which to instantiate applications providing the embedded distributed real-time control policies. Further discussion of such a platform in the form of an EOS, is provided herein.

[0096] Before discussing the EOS and other components used in certain methods and systems of the invention, the behavior of federated enterprise systems is discussed. Fig. 7 provides an illustration of system behavior in the form of a graph 65. The graph 65 includes nodes 67 (vertices) which are processing elements (processes). The nodes 67 are interconnected by arcs 70 (communications links). A set of graphs can be used to model a given system and to represent behaviors of interest (i.e., scenarios) under various operating regimes (i.e., requirements), and therefore form the basis for defining the syntax and semantics of the system itself.

The example provided in Fig. 7 represents a particular (instantaneous) configuration of nodes and links for three interconnected systems 72, 73, and 74 (labeled A, B, and C). Enterprise-to-enterprise threads 76 connect the systems 72, 73, and 74.

[0097] A given node 67 (also identified as "x" in Fig. 7) may in fact be a member of two or more such enterprises (graphs) simultaneously. In such cases, the semantics of node "x" vary in relation to each system in which it is embedded. This leads to potential ambiguity in interpreting its behavior by a single thread, or by, for example, computational intelligence ("CI") analysis that is not "context sensitive." This issue is more important when systems, under the pressures of their local dynamics, evolve or adapt. If node "x" were an element of such an adaptive subsystem, an observer of its behaviors in each of the systems it populates would see different measures of its performance, and the differences would fluctuate over time and operating conditions. As will be discussed in greater detail, embodiments of the invention provide mechanisms for measuring performance for such elements.

[0098] A VPU 80 is shown in Fig. 8. The VPU 80 participates in an asset chain 82 (shown along a vertical axis) and serves its investors or suppliers of assets. The VPU 80 also participates in a supply chain 84 (shown along a horizontal axis) and serves its customers or consumers of products or services. The asset chain 82 and supply chain 84 form a grid or lattice 85. A given enterprise may include one or more VPUs and Fig. 8 illustrates a number of VPUs (86, 88, 90, and 92) connected to the VPU 80 through either the asset chain 82 or supply chain 84.

[0099] Each VPU (80, 86, 88, 90, and 92) supports its asset and supply chains through four full-duplex ports or information inlets and outlets. The ports are illustrated as eight communications ports, but could be implemented in a variety of ways. Each VPU could be readily constructed as a software object and the ports could be implemented as methods or other similar mechanisms (e.g., procedure or function calls and returns) that provide a mechanism of communicating information to and from devices or logical constructs. The ports used in one embodiment are defined in Table 1 below. Investors provide assets at port a_i (assets-in) that subsequently

yield investment returns on port r_o (returns-out). Customers provide demands for goods or services on port d_i (demand-in) that are fulfilled on port s_o (supply-out).

Table 1

Port ID	Name	Port Function (Protocol)
ao	Assets_out	Send to subordinate requested production assets
ai	Assets_in	Receive from superior requested production assets
ro	Returns_out	Send to superior returns generated from assets deployed
ri	Returns_in	Receive from subordinate returns from assets deployed
do	Demand_out	Send to supplier demand for "raw material"
di	Demand_in	Receive from client demand for supply
so	Supply_out	Send to client fulfillment of demand
si	Supply_in	Receive from supplier fulfillment of demand

[0100] Although, not shown, it is to be understood that each port may service multiple "connections," supporting the VPU's simultaneous participation in several supply and asset chains. The multiplicity of connections on an input port is called its "fan-in" and on an output port its "fan-out."

[0101] A VPU supports two subsidiary channels, one for subordinate VPUs (typically for intellectual property generation), and one for supplier VPUs (typically for material stocks.) Subordinate VPUs are allocated investment assets on port a_o (assets-out) that generate returns on port r_i (returns-in). Supplier VPUs receive their demands on port d_o (demand-out) and return their production on port s_i (supply-in).

[0102] Each VPU is uniquely identified (i.e., named) by its location vertically and horizontally in the grid 85, as is illustrated using coordinate subscripts k and l. Thus, $VPU_{k,l}$ (or 80) is subservient in the asset chain 82 to $VPU_{k,l+1}$, (or 86) and is a supplier to $VPU_{k+1,l}$ (or 88) in the supply chain 84. A VPU acts as a virtual machine (an "actor") whose behavior, governed by a "program," is the continuous execution of the system's value proposition(s). Value propositions comprise the VPU's logic that governs actions ("methods") that carry out the strategic, operational, and tactical goals and objectives that add value to the environment (commons) within which the system functions.

[0103] Value propositions may take the form of the statement

if <condition> then <action$_1$>, else <action$_2$>.

Where "action" defines a step in a value production sequence, and "condition" tests for the presence or absence of required production assets, consumable resources or production side effects.

[0104] VPUs may also process (i.e., produce and respond to) asynchronous "events." The constructs for event posting and notification are, respectively

post: signal <event>

catch: on <event> do <action>

Having at least one VPU present in multiple systems defines a family of systems, sharing a set of value propositions carried in the "DNA" or fundamental structures of the common VPUs.

[0105] Fig. 9 illustrates how a VPU 100 interacts in a system of systems. The VPU 100 is the most superior unit in system C, is the most subordinate unit in system B, and is an intermediate unit in system A.

[0106] The generalized and symmetric structure of the VPU model allows for the creation of arbitrarily complex webs of relationships. The lattice or grid 85 may be configured to model VPUs representing levels (L_0-L_n) of the vertical asset chain 82, and multiple levels in the supply chain 84.

[0107] The connections among VPUs are assumed to be dynamic, meaning that they are established and broken as the enterprise or federation operates. To support this dynamic feature, a transport system 130 (Fig. 10) capable of binding the ports of VPUs is provided. The transport system may be a mechanism similar to a telephone exchange or Internet router. With the services of the transport system 130, the asset and supply chains may be interconnected and dynamics may be investigated.

[0108] Figs. 11 and 13, illustrate two examples of VPU production functions or models, a model 135 representing a generalized mass-flow production system, and a model 140 representing a generalized micro-economic financial system.

[0109] The model 135 defines asset chain and supply chain transformation functions. In the asset chain dimension, the flows are governed by four internal functions (π_{aa}, π_{ar}, π_{ra}, and π_{rr}) and two internal routing parameters (δ and γ). The interpretation of these functions and parameters, as well as those for the supply chain, are defined in Table 2.

Table 2

	Asset Chain Definitions
π_{aa}	Asset allocation function to subordinate VPUs
π_{ar}	Asset allocation function to generate additional internal capacity
π_{ra}	Asset return allocation function to generate reinvestment in subordinate VPUs
π_{rr}	Asset return allocation function to generate additional internal capacity
δ	Asset allocation proportioning parameter
γ	Asset return proportioning parameter
	Supply Chain Definitions
ϕ_{dd}	Demand allocation function to generate supplier (server) demand
ϕ_{ds}	Demand allocation function to provide fulfillment for client demand
ϕ_{sd}	Demand fulfillment allocation function to generate additional supplier demand
ϕ_{ss}	Demand fulfillment allocation function to provide fulfillment of client demand
α	Demand allocation proportioning parameter
β	Demand fulfillment proportioning parameter

[0110] The model 135 is oriented towards an abstract enterprise concerned with value production and the flow of command and control threads. The model 135 defines a set of transport functions 142 and 144 (Fig. 12).

[0111] Assessing system behavior requires a set of metrics. Fig. 14 illustrates six performance metrics 160 (potentiality), 162 (capability), 164 (actuality), 166 (latency), 168 (productivity), and 170 (performance). The performance metrics 160-170 are capable of evaluating the performance of value production processes in an application process-neutral way. The six metrics comprise three basic measures (160, 162, and 164) for evaluating system capacity, and three derived measures for evaluating system achievement (166, 168, and 170) towards its goals and objectives.

[0112] Potentiality (160 or P) is a system's desired capacity to do work. Potentiality is what a system ought to be able to do, all things being equal. There is,

however, only a subset of the system's overall potential that is actually available for value generation, this being its capability (162 or C). Capability is what a system could do if it fully utilizes its resources. However, at any given time a system, possibly due to failures, lack of raw materials, poor processes, or labor disputes, will likely perform below its full capability. This level of performance is the system's actuality (164 or A).

[0113] In order to compare the performance of federations of independent or partially independent processes that may have widely divergent potentials, it is necessary to normalize the metrics (160, 162, and 164). In Fig. 14, the values of P, C and A have been normalized to a relative value of P (P_{rel}) within each system, where

$$P = P_{rel}/P_{rel} = 1.00$$

$$C = C_{rel}/P_{rel} = 0.75$$

$$A = A_{rel}/P_{rel} = 0.35$$

[0114] As a consequence of normalization, capability 162 and actuality 164 become percentages of potentiality 160. The derived measures for evaluating system achievement (metrics 166, 168, and 170) are derived from the normalized metrics.

[0115] The ratio of capability to potentiality is the system's latency 166 ($\lambda = C/P$), representing the amount of unused capacity (latent potential, or unutilized resources). Through operational planning and process improvements, a system may be able to raise its capability to gain incremental improvements in performance while remaining within its design (i.e., architectural) constraints.

[0116] The ratio of actuality (A) to capability (C) represents the available but unused capacity in the system. It is a measure of the system's productivity 168 ($\rho = A/C$). Through tactical programing, existing resources (assets) can be made more productive.

[0117] The ratio of actuality (A) to potentiality (P) represents the system's absolute performance 170 ($\pi = A/P$). Alternatively, performance may be computed

from latency and productivity by the relationship $\pi = \lambda * \rho$. In the example of Fig. 14, the following achievement indices are derived from the basic metrics P, C and A: $\lambda = .75$, $\rho = .47$, $\pi = .35$.

[0118] Two alternate scenarios 175 and 177 are also presented in Fig. 14. In scenario 175, an increase in A of .07 (A* = .42) results in a significant increase in productivity 168, from $\rho = .47$ to $\rho^* = .56$. There is a corresponding rise in performance 170, from $\pi = .35$ to $\pi^* = .42$.

[0119] In scenario 177, actuality 164 remains at .35 while capability 162 is raised from C = .75 to C* = .85. There is an expected drop in productivity 168, from $\rho = .47$ to $\rho^* = .41$, and there is a somewhat counter-intuitive drop in performance 170, from $\pi = .35$ to $\pi^* = .31$. The alternative scenario 177 illustrates that problems can occur if a balance is not achieved and maintained between potentiality 160, capability 162, and actuality 164.

[0120] In addition to the scenarios 175 and 177, it is possible that commitments may be made when, in fact, there are too few available resources (capability) to meet the commitments. In this "over committed" state, the operational elements of the system must reprioritize the work; letting some commitments suffer delays. Alternatively, operations must scramble to put on additional capacity. This situation, typically driven by random fluctuations in demand and unplanned failures of key resources, requires that P>C. As a consequence, $\lambda < 1$, and the system control problem becomes focused on maximizing productivity of available resources, ρ. While this may seem logical, the fact remains that planning in general, and capacity planning in particular, is critical to the regulation of any system, but especially for systems that must grow and adapt to remain viable.

[0121] A mechanism 180 that may be used to provide such regulation is illustrated in Fig. 15. The mechanism 180 includes a feedback control device 182 augmented with autonomic (or self-regulatory) device 184. Fig. 16 illustrates an exemplary viable system controller 190 for a VPU. There are two basic processes to control, an asset chain process (ACP[k,l]) or 191A and a supply chain process

(SCP[k,l]) 191S. Another way of looking at the controller is to think of it as a single business unit (BU) or 192 (Fig. 16), structured in a way that supports designed or engineered control. Fig. 16 peers into the business unit 192 to uncover its VPU 193, as defined by its structure as a viable system. Fig. 16 identifies the essential internal objects that provide the dynamic behavior of value production as defined above. The asset chain process ("ACP") 191A and supply chain process ("SCP") 191S are responsible for engaging neighboring vertical and horizontal VPUs in their respective chains. ACP 191A and SCP 191S are two echelon 1 ("E1") components and contain the fundamental operations of the viable system, the VPU's *implementation*. The ACP 191A and SCP 191S do not include higher level management functionality. Management is considered a service to fundamental processes and is included as part of the supervisory, operational & planning, and regulatory functions at echelon 2 ("E2"), echelon 3 ("E3"), echelon 4 ("E4"), and echelon 5 ("E5").

[0122] With respect to generally accepted accounting practices ("GAAP"), a BU's ACP activities are recorded on its balance sheet 140A (Fig. 13), and its SCP activities are summarized on its income statement 140B. The tradeoffs required in the management of these two views are the responsibility of higher echelons in the model or, in other terms, the E5-E4-E3 management team. Thus, in the example shown in Fig. 16, an E1_ACP management component 194 ("E1_ACP_Sup") is focused on acquiring and deploying assets (e.g., infrastructure development), and an E1_SCP management component 195 ("E1_SCP_Sup") is focused on the sale and delivery of products and services.

[0123] In total, there are eleven components identified in Fig. 16 comprising the VPU 193. They are the ACP 191A, the SCP 191S, the E1_ACP_Sup 194, the E1_SCP_Sup 195, an E2_ACP_Reg 196A, and E2_SCP_Reg 196B, an E2_VPU_Reg 196C, an E3_VPU_Ops 197A, E3_VPU_Audit 197B, an E4_VPU_Dev 198, and E5_VPU_Sup 199. The objects 191-199 are linked by network or grid connections, allowing the objects to be distributed in whatever manner is suitable to the application at hand. A variety of message formats and protocols may be used to support communications between the objects, but such formats and protocols may be based on

standards such as those found at www.w3.org, www.jcp.org, www.omg.org, www.ietf.org, and www.rosetta.org.

[0124] The regulatory loops for the ACP and SCP activities (e.g., E1_ACP, to E2_ACP_Reg, to E1_ACP_Sup, back to E1_ACP) illustrate the need for a protocol machine and set of measurement and control messages. In some embodiments of the invention, the control loop may be patterned after classical feedback controllers.

[0125] For predictability and stability reasons, regulatory loop timing requirements must be specifiable and consistent within a viable system model ("VSM"). The VSM within the E1_ACP process, for example, must adhere to timing requirements that do not conflict with those of its encapsulating E2_ACP_Reg. And those must not, in turn, conflict with the E3_VPU_Ops operations loop.

[0126] It is preferable that the E2_VPU_Reg regulator be able to prioritize and preempt operations in the E1_ACP and E1_SCP VSMs. This capability requires that policies and mechanisms exist to support coordination and synchronization, assisting E2_VPU_Reg with its role in damping oscillatory behaviors as well as avoiding deadlocked behaviors that may result from contention over shared resources.

[0127] In addition, because the VSM is recursively defined, the object interfaces, protocols, and message syntax should scale, and not be level specific. Fig. 5 illustrates expanding and coupling VPUs (that represent the business units 41, 42, and 43) within a VPU defining the BA 40. Fig. 5 shows the BU VSMs rotated 45 degrees in order to fit them into the BA VSM. Likewise, "under a microscope" we'd see the asset and supply chain VSMs within the BU's. And in a similar fashion, zooming outward, we'd find the BA embedded in a corporate VSM structure that may contain other BAs. This scoping applies from the lowest enterprise VSM levels (e.g., a manufacturing cell within a factory within a BU) up through alliances among corporations, entire vertical market segments, and national and global markets.

[0128] The model provided in Figs. 5 and 16 may be implemented in a variety of manners, including in software. As noted above, there are eleven key components (i.e., classes) in the VPU object model, including E1_ACP, E1_SCP, E1_ACP_Sup,

E1_SCP_Sup, E2_ACP_Reg, E2_SCP_Reg, E2_VPU_Reg, E3_VPU_Ops, E3_VPU_Audit, E4_VPU_Dev, and E5_VPU_Sup. In one embodiment, these components can define classes in an object-oriented programming language such as Java. The services of each class, as implemented in an exemplary embodiment are defined below.

[0129] The ACP may be an enterprise object class (e.g., "enterprise java bean") defined by

- An *asset production model* deployed in a VPU
- A set of transactions (messages and protocols) that interface the model to an enterprise's underlying financial (e.g., ERP) systems
- A set of metrics (e.g., Six Sigma) for auditing the performance of the ACP
- A set of trading interfaces to the relevant financial markets

[0130] The SCP may be an enterprise object class defined by

- A *supply production model* for products produced in and exchanged by the VPU
- A set of transactions (messages and protocols) that interface the model to an enterprise's underlying manufacturing (e.g., MRP or project management) systems
- A set of metrics (e.g., Six Sigma) for auditing the performance of the SCP
- A set of trading interfaces to the relevant product markets

[0131] The asset chain supervisor E1_ACP_Sup provides direct administrative controls over the E1_ACP, and includes such services as

- Receive, interpret and execute commands from E3_VPU_Ops
- Formulated and send status of E1_ACP to E3_VPU_Ops
- Develop normative execution plans for E1 operations
- Supervise the regulatory actions of E2_ACP (e.g., operational set-point controls)

[0132] The supply chain supervisor E1_SCP_Sup provides direct administrative control of E1_SCP, and includes such services as

- Receive, interpret and execute commands from E3_VPU_Ops
- Formulate and send status of E1_SCP operations to E3_VPU_Ops

- Develop normative execution plans for E1_SCP operations
- Supervise the regulatory actions of E2_SCP (e.g., operational set-point controls)

[0133] The asset chain regulator E2_ACP_Reg provides the feedback controls that support reflexive (autonomic) controls over behavior of the asset chain process. Through E2_ACP_Reg's four interfaces,

- It senses and responds to activities in the E1_ACP process
- It couples the normative supervisory control functions of E1_ACP_Sup to the E1_ACP production process
- It connects to and supports the regulatory functions of the VPU operations level through E2_VPU_Reg
- It connects to the supply chain regulator E2_SCP_Reg in order to coordinate with actions of the E1_SCP process

As such, the asset chain regulator E2_ACP_Reg participates in four control loops, and is an important element in the VPU's ability to attain and sustain homeostasis.

[0134] Like its E2_ACP_Reg counterpart, the supply chain regulator E2_SCP_Reg provides feedback controls that support reflexive (autonomic) controls over behavior of the asset chain process. Through E2_SCP_Reg's four interfaces,

- It directly senses and responds to activities in the E1_SCP process
- It couples the normative supervisory control functions of E1_SCP_Sup to the E1_SCP production process
- It connects to and supports the regulatory functions of the VPU operations level through E2_VPU_Reg
- It connects to the supply chain regulator E2_ACP_Reg in order to coordinate with actions of the E1_ACP process

[0135] The E2_VPU_Reg encapsulates and coordinates the behaviors of value and supply chain production. It is responsible for managing the resources and synchronizing the events that are required for this role. A key function in support of that responsibility is coordination of the two process regulators, E2_ACP_Reg and E2_SCP_Reg. This role is performed by E2_VPU_Reg, whose services include

- Provide real-time status to E3_VPU_Ops relative to E1 activities and VPU objectives
- Accept "override" commands (e.g., set-point changes) from E3_VPU_Ops in response to E1 exceptions
- Balance the real-time demands of the E1 processes against one another according to E3 policy

[0136] The E3_VPU_Ops, as a proxy for a responsive (i.e., reflexive and adaptive) value production entity within an enterprise, operates according to plans that undergo constant revision. Such plans are the result of combining history, operational pragmatics, current objectives, resource constraints, and incremental developmental improvements. It falls to the operations directorate to continually assess and rationalize these aspects and produce executable programs for the E1-level directorates. To do so E3_VPU_Ops requires the ability to independently assess current activities in its E1 processes (E3_VPU_Audit), react to real-time E1 events (E2_VPU_Reg), and to participate in the planning of incremental change (E4_VPU_Dev). Through these interfaces, E3_VPU_Ops provides the following services.

- Continuously receive, filter interpret and respond, through the exception-reporting services of E2_VPU_Reg, the real-time behavior of E1 production systems
- Continuously interrogate, interpret, filter and report to E4-E5, through the auditing services of E3_VPU Audit, the current status of specific E1 activities
- Periodically develop, revise, deploy (to E1_SCP_Sup and E1_ACP_Sup) and monitor tactical operating plans received from E5_VPU_Sup and E4_VPU_Dev that achieve the [typically, near-term] objectives of the VPU
- Support incremental "reprogramming" of the VPU in order to implement innovations provided by E4_VPU_Dev and mission directives from E5_VPU_Sup.

[0137] The E3_VPU_Audit assesses the current state of a system using a set of uniform and consistently applied metrics. A main role of the E3_VPU_Audit is to establish a uniform frame of reference for E3_VPU_Ops in carrying out its

responsibility for achieving milestones in its operating plans. As such, the E3_VPU_Audit class provides

- A defined set of operational performance metrics (e.g., Six Sigma)
- A defined set of measurement methods
- A schedule for performing the methods within the E1 domains
- A reporting protocol for communicating the results to E3_VPU_Ops and to the E1_ACP and E1_SCP processes.

[0138] The E4_VPU_Dev provides a mechanism to develop plans of adjusting to the changes suggested by measurements made by the E3_VPU_Audit. Thus, the E4_VPU_Dev is configured to address three dimensions of VPU behavior:

- Its vision and mission (i.e., the objectives that result from its E5 belief system, and how it manages innovation through its E4 development programs)
- Its core capabilities as defined in its E1_SCP and E1_ACP processes
- Its infrastructure (i.e., how it functions through the services of E2 and E3)

[0139] These requirements are supported by E4_VPU_Dev services that include

- VPU modeling and simulation
- Competitive environment measurement and assessment
- Recommendations to E5 on adjustments to vision and mission objectives
- Recommendations to E3 on operational improvements (e.g., infrastructure development)
- Recommendations to E1 on product and process changes

[0140] The E5_VPU_Sup provides a superior authority or point of accountability. The responsibilities of the E5_VPU_Sup include

- Establishing and maintaining the identity, vision, and mission objectives of the VPU (i.e., its reason for existence)
- Deriving and enforcing the policies (doctrines) that derive from the mission objectives
- Representing the VPU in the affairs of the metasystem in which it is an element (recursively defined)

[0141] The E5_VPU_Sup (together with the E4_VPU-Dev and E4_VPU_Ops) is, at the next higher level, the E1_x_Sup, where x is defined in the context of that level.

[0142] Having described details of the application of the VPU construct to components of an enterprise, the issue of performance measurement is again taken up.

[0143] The controller 190 may be modified to include performance measurement capabilities, as is best illustrated in Fig. 17A. As shown in Fig. 17A, performance of a system designed to measure and control a process is measured using the metrics introduced above (potentiality 160, capability 162, actuality 164, latency 166, productivity 168, and performance 170). In addition, an audit process or control 200 is added. The audit control 200 provides a mechanism for establishing a basis for value judgment. It serves, at least in part, the function of an "open market" valuation among independent traders in goods and services.

[0144] As can be seen by reference to Figs. 17A and 17B, measures of actual performance are provided by the E1 processes through an actuality measurement tool 202; measures of capability are provided by the E3 operations directorate through a capability measurement tool 204; and measures of VPU potentiality are provided by the E4 planning function through a potentiality measurement tool 206. These three sets of measures form the basis for calculating the VPU's latency, productivity, and performance indices. Doctrine is generated at the E5 level and a doctrine generation tool 207 may be used to assist in the generation of the doctrine. Performance, latency, and productivity, as derivatives, may be calculated, as discussed above with a performance metrics engine 208. In addition to the six metrics (160, 162, 164, 166, 168, and 170) qualified results of the (E3*) audit process 200 are also available for export or output to management of higher echelons.

[0145] Putting these elements together, and recognizing the nested (recursive) natures of C2 lattices, results in a performance or production measurement framework ("PMF") 210 (Fig.18A).

[0146] The PMF 210 include many features already discussed and additional PMF services, including performance reporting 212, performance analysis 214, and

performance data acquisition 216. The PMF 210 also includes an interconnect structure 220, which may be the same grid or network infrastructure used by the enterprise to conduct its normal activities. In this respect, the PMF 210 is itself a service-oriented environment.

[0147] Fig. 18B illustrates a generalized PMF model 230 specialized for the VPU controller 190. The PMF model 230 includes two echelon 1 processes 234 and 238 and the means for their performance assessment in the form of ACP and SCP VPUs, 239 and 240 (the concepts of which were described above).

[0148] Fig 19. Illustrates a management or command and control system 245. Data from components of the system (for example, the first three echelons of VPU's and other systems such as ERP and CRM systems) is transferred through an interface module 246. Operation commands are also directed out to lower echelons through the modules 246. The raw information received in the interface modules 246 is delivered to a data acquisition service 247 that may store raw information in a database 247DB. The data acquisition service 247 also delivers the raw information to one or more filters, represented by a filter 248. Information processed in the filter 248 may be store in a database 248DB and delivered to other components of the system 245, including a performance measurement engine 249 and an alarms and events engine 250. The performance measurement engine 249 performs the functions described with respect to Figs. 14, 17A and 17B to generate performance metrics. Performance metrics from the performance measurement engine 249 are delivered to a history engine 251. The history engine 251 receives input from the alarms and events engine 250 and stores performance metrics in a history database 251DB. The history engine 251 is coupled to a report generator 252, which produces reports based on the performance metrics and historical information received from the history engine 251.

[0149] The alarms and events engine 250, history engine 251, and reports generator 252 are coupled to a display generator 253, which takes information from these three sources and produces output that may be presented on a CRT, LCD, printer, or similar output device. For example, the display generator 253 may create graphical interfaces (exemplified by output screens or interfaces 253VOA, 253VOB,

and 253VOC). Various displays may also be shared on a wide-area basis (as is represented by the display 253VOG (video output global)).

[0150] The interfaces 253VOA-253VOC are also input mechanism through which individuals controlling the system 245 may enter inputs or commands. The commands are routed to a command parser 254, which separates what are referred to as bridge commands (commands meant for individual components or levels within the system 245, as represented by the L0 through L5 constructs) from operational commands (commands that affect the entire system). Bridge commands are directed to a router 255 which directs the commands to the appropriate level (or, to take a more granular perspective, to a VPU, or echelon of a VPU, or process of a VPU). Operational commands are directed to the alarms and events engine 250.

[0151] Commands from the bridge interfaces 253VOA, 253VOB, and 253VOC may also be directed to a modeler 256 such that individuals in charge of the system 245 may create models of future behavior of enterprise components (such as a business unit) as needed. Information from each model created is delivered to the performance measurement engine 249 so that an appropriate measurement service for the newly modeled components may be established. In addition, information from the modeler 256 is delivered to an icon engine 257 and associated database 257DB to establish appropriate graphical tools for the newly modeled components. Information from the icon engine is shared with the display generator so that appropriate visual information may be displayed.

[0152] As noted above, the domain of enterprise management is complex and subject to broad interpretation. Interpretations depend on application context, the more obvious being military, commercial, public health, and government. Interpretations are further complicated by regional and cultural biases in the application of command and control by various practitioners. Business personnel operate differently than military officers, both of whom use different protocols than physicians in medical practices or supervisors on large construction projects. There are, however, similar objectives and practices across these domains that are relevant to the specification of distributed real-time enterprise. To codify the common features

requires a lexicon from which a formal syntax and a semantic are derived in the form of an EOS.

[0153] Fig. 20 illustrates an EOS 260. The EOS 260 provides elements and features to support distributed real-time command and control applications. At a base or network interface layer or level 261 of the EOS 260 are interfaces 262-274 to run-time services (historian, alarms & events, messaging, security, diagnostics, scheduling, time, etc.) (Generically enterprise virtual machine ("EVM") services 275) that support predictability and timeliness. The base or network interface level 261 also includes interfaces to grid-connected applications such as the an enterprise resource planning ("ERP") interface 280, a manufacturing resource planning ("MRP") interface 282, a distributed control systems ("DCS") interface 284, a document management system ("DMS") interface 286, a customer relationship management ("CRM") interface 288, an engineering design system ("EDS") interface 290, and a CMS interface 292 (generically, enterprise application interfaces ("EAIs") 294). Of course, other interfaces (not shown) could be included.

[0154] Fig. 21 illustrates commercial enterprise applications categories, or suites, that may be coupled or interfaced to the EOS 260. (Military applications would populate an equivalent map for a given military branch.) The location and coloring emphasizes relative positions in the C2 automation space and the fact that these suites tend to be isolated, of different manufacture, and from different vendors. The applications may include those mentioned above ERP, CRM, etc.

[0155] Referring back to Fig. 20, the EOS 260 includes a second or performance measurement layer or level 300 having value production processes ("EVPs") of a virtual organization ("VO"). In the exemplary EOS, production processes include production units 302, production areas 304, business units 306, business area 308, and corporate governance 310. Above the EVP or second level 300 is a third or process control layer or level 311 of enterprise process controls ("EPCs") that govern autonomic behaviors. In the EOS 260, the ECP include a processes service 312, a control service 314, a coordination service 316, a development service 318, and a supervision service 320. The EOS 260 also includes a fourth layer or level 330 that

provides enterprise management interfaces ("EMIs"). The EMIs includes a stationary bridge 332, a mobile bridge 334, and authentication service 336, and a reporting engine or service 338.

[0156] The EMIs of level 330 provide tools or services to facilitate decentralized management of complex extended enterprises. Fig. 22 illustrates a bridge display 348 with multiple EMIs that may be generated by a system based on the teachings herein. The display 348 includes a status display 350 of a VPU 355. The display 350 contains three primary elements: performance meters 360 (on the left); quality, capability, and yield meters 362 (on the right); and financial graphics 364 (in the center). In the example shown, the performance meters 360 display latency, productivity, and performance. The quality, capability, and yield meters display process quality, in this case its variation (sigma), capability limits, and throughput yield of the VPU 355. The financial graphics 364 display more traditional financial measures of performance, including sales volume, return on invested capital, headcount, cost of goods sold, assets deployed, etc.

[0157] To the left and right hand edges of the bridge display 348 are buttons 368 (e.g., links) for activating web pages or other GUIs or content associated with two distinct sets of VPU controls – portals for VPU constituents, and portals for VPU echelon controls. Fig 22 illustrates four portals: an investor portal 370, a customer portal 372, a supplier portal 374, and a subordinate portal 376. Five echelon controllers are also illustrated: a strategy controller 378, a development controller 380, an operations controller 382, a supervision controller 384, and a process controller 386.

[0158] The bridge display 348 includes means for identifying and selecting a VPU (or focus) in the form of a dialog box 390; where the name or address of the VPU of interest may be input. Thus, the bridge 348 provides a tool through which it is possible to visit the key production processes of an organization (or virtual organization), in a logical order, to view performance measures, to identify value chain behavior, and issue commands. Proactive commands initiate behaviors that will potentially affect connected VPUs. In the opposite direction, asynchronous alarms

and events (e.g., signaling completions, failures, resource limits, etc.) generated within VPUs will, with some filtering, flow towards the command centers, ultimately to arrive on one or more portal pages or bridges for the attention of relevant controllers (managers). In both directions, timeliness and predictability are critical requirements for achieving and sustaining enterprise viability.

[0159] The EOS model as illustrated in Fig. 20 recognizes six organizational levels (L_0-L_5), each populated by one or more VPUs, and each VPU governed by a five-echelon controller and simultaneously serving two value chains through four interfaces. This aspect of the model is illustrated in Fig. 23. Thus, an EOS representing an enterprise can have numerous dimensions. The number of dimensions is determined by the following formula:

[0160] No. of dimensions = (no. of processes/no. of VPUs * no. of VPUs/level * 5 controllers/VPU * 6 levels/enterprise).

[0161] As should be apparent, the number of dimensions could be very, very large for an enterprise or federation of enterprises with hundreds or thousands of VPUs. The EMIs (for example, bridge display 348) provide a tool that allows navigation and control of such a large space.

[0162] It should also be noted that the concepts of superior, subordinate, server, and client VPU portals will change in context, style and function for military C2 applications, and that the context will further vary depending on which military branch the EOS is being applied to.

[0163] Fig. 23 also illustrates how continuous process improvement can be implemented using an EOS (such as the EOS 260). Fig. 23 depicts three VPUs 402, 404, and 406. Each VPU is associated with a PMI 412, 414, and 416, respectively. Each PMI displays each VPU's performance characteristics. The PMI 412 for VPU 402 depicts the performance seen by its E4 controller of its level 1 activity sometime in the past. The PMI 414 for VPU 404 depicts the performance seen by its E2 regulatory controller of its level 4 activities at this moment. And the PMI 416 for

VPU 406 depicts the performance that might be seen in the future by its operational controller at level 2.

[0164] To help further elucidate the EOS 260 and other features and aspects of embodiments of the invention, Figs. 24 – 26 illustrate graphical summaries of certain components and elements already described. Fig. 24 illustrates major elements of the EOS 260 atop a grid or network 450. The grid or network 450 represents the many systems or inputs that supply information to embodiments of the invention (e.g., the ERP, MRP, DCS, CRM, EDS, CMS, and other systems mentioned above). At the base of the EOS 260 are the virtual machine services (EVM services) and the grid-level I/O subsystem (EAIs) of level 261 providing access to web attached (for example, via extensible mark up language ("XML"), Web Services Description Language ("WSDL"), Simple Object Access Protocol ("SOAP"), Universal Description, Discovery and Integration ("UDDI"), or other tools or interfaces) enterprise applications as summarized in Fig. 21. Within each ring are value production class objects or VPUs that define the core production processes of the extended or virtual enterprise or VO of level 300. The respective echelon controllers or EPCs of level 311, in turn, regulate these objects as they participate in their asset and value chain computations. The command and control environment is presented to end-users through the enterprise management interfaces or EMIs of level 330.

[0165] Fig. 25 summarizes key elements of the enterprise performance management user interface. The echelon 5-4-3 interfaces are concerned primarily with the acquisition and management assets, particularly their performance relative to the federation's need for producing value. Fig. 25 includes a controller page with selectors along the top, a VPU navigator on the left, and a three-panel performance display in the center.

[0166] A more specific example of a PMI (PMI 500) directed to an echelon 5 supervisory controller (e.g., corporate officer, theater commander, etc.) is presented in Fig. 26. The PMI 500 is financially oriented and depicts the real-time financial status of an enterprise operation. The PMI 500 includes a page 502 that is an idealized

summary of the profit and loss statement, the balance sheet, and snap-shots of the roll-up of all of the enterprise's embedded systems.

[0167] As should be apparent to one of ordinary skill in the art, the systems shown in the figures are models of what actual systems might be like. Many of the modules and logical structures described are capable of being implemented in software executed by a microprocessor or a similar device or of being implemented in hardware using a variety of components including, for example, application specific integrated circuits ("ASICs"). Thus, the claims should not be limited to any specific hardware or software implementation or combination of software or hardware.

[0168] Various features and advantages of the invention are set forth in the following claims.